DATA PAPER

Constructing and Cleaning Identity Graphs in the LOD Cloud

Joe Raad[1†], Wouter Beek[1], Frank van Harmelen[1], Jan Wielemaker[1], Nathalie Pernelle[2] & Fatiha Saïs[2]

[1]Deptartment of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1085, 1081 HV Amsterdam, The Netherlands
[2]Computer Science Research Laboratory (LRI) of the University Paris Sud, French National Centre for Scientific Research, Paris Saclay University, 91190 Saint-Aubin, France

Keywords: Linked Open Data; Identity; Quality; Reasoning

Citation: J. Raad, W. Beek, F. van Harmelen, J. Wielemaker, N. Pernelle & F. Saïs. Constructing and cleaning identity graphs in the LOD cloud. Data Intelligence 2(2020), 323–352. doi: 10.1162/dint_a_00057
Received: August 31, 2019; Revised: September 20, 2019; Accepted: September 30, 2019

ABSTRACT

In the absence of a central naming authority on the Semantic Web, it is common for different data sets to refer to the same thing by different names. Whenever multiple names are used to denote the same thing, owl:sameAs statements are needed in order to link the data and foster reuse. Studies that date back as far as 2009, observed that the owl:sameAs property is sometimes used incorrectly. In our previous work, we presented an identity graph containing over 500 million explicit and 35 billion implied owl:sameAs statements, and presented a scalable approach for automatically calculating an error degree for each identity statement. In this paper, we generate subgraphs of the overall identity graph that correspond to certain error degrees. We show that even though the Semantic Web contains many erroneous owl:sameAs statements, it is still possible to use Semantic Web data while at the same time minimizing the adverse effects of misusing owl:sameAs.

† Corresponding author: Joe Raad (E-mail: j.raad@vu.nl; ORCID: 0000-0002-7891-7738).

1. INTRODUCTION

As the Web of Data continues to grow, more and more large data sets – covering a wide range of topics and domains – are being added to the Linked Open Data (LOD) Cloud. It is inevitable that different data sets, most of which are developed independently of one another, will come to describe (aspects of) the same thing, but will do so by referring to that thing by different names. This situation is not accidental: it is a defining characteristic of the (Semantic) Web that there is no central naming authority that is able to enforce a Unique Name Assumption (UNA). Thanks to identity links, data sets that have been constructed independently of one another are still able to make use of each other's information. As a consequence, identity link detection, i.e. the ability to determine – with a certain degree of confidence – that two different names in fact denote the same thing, is not a mere luxury but is essential for Linked Data to work. The most common property that is used for interlinking data on the Web is owl:sameAs. An RDF statement of the form "x owl:sameAs y" indicates that every property attributed to x must also be attributed to y, and *vice versa*.

Over time, an increasing number of Semantic Web studies have shown that the identity predicate is used incorrectly for various reasons (e.g. heuristic entity resolution techniques, lack of suitable alternatives for owl:sameAs, context-independent classical semantics). This misuse has resulted in the presence of a number of incorrect owl:sameAs statements in the LOD Cloud, with some studies estimating this number to be around 2.8% [1] or 4% [2], whilst other studies suggest that possibly one out of five owl:sameAs in the Web is erroneous [3].

In order to limit the problem of identity links in the Semantic Web, a number of different attempts have emerged over the recent years (for a more exhaustive background, we refer the reader to [4]). In 2007, the authors of [5] proposed the centralised entity naming system OKKAM as a way to encourage the reuse of existing names. This centralised solution did not receive uptake in the wider Semantic Web community. This is not entirely unexpected, since the introduction of a centralised naming authority amounts to a strong departure from the original (Semantic) Web ideology, while at the same time requiring significant changes to the use and interchange of Linked Data in practice. Other approaches tried to limit the Semantic Web identity problem by providing centralised access to identity statements that are published in a decentralised way. Such centralised identity services allow Linked Data consumers to make an informed decision regarding the quality of identity statements they encounter. Although such services can in theory be useful for addressing the identity problem, current services suffer from semantic [6] and/or coverage [7] limitations. Firstly, the 'identity bundles' that the Consistent Reference Service [6] provides access to, are the result of the equivalence closure over owl:sameAs statements in addition to statements with different or no semantics. For example, umbel:isLike statements denote similarity instead of identity and are symmetric but not transitive; skos:exactMatch statements are symmetric and transitive, but indicate "a high degree of confidence that the concepts can be used interchangeably across a wide range of information retrieval applications," which is semantically very different from the notion of identity. As a result, the semantics of the closure that is calculated over this variety of statements is unclear. On the other hand, LODsyndesis [7] is a

co-reference service based solely on owl:sameAs statements, but the number of triples it covers is an order of magnitude smaller than the ones we present in this work.

Besides the above described services, various approaches have been proposed for detecting erroneous identity statements, based on the similarity of textual descriptions associated to a pair of linked names [8], Unique Name Assumption (UNA) violations [9, 10], logical inconsistencies [1, 11], network metrics [12], and crowdsourcing [13]. However, existing approaches either do not scale in order to be applied to the LOD Cloud as a whole, or they make assumptions about the data that may be valid in some data sets but not in others. For example, in the LOD Cloud not all names have textual descriptions, many data sets do not include vocabulary mappings, or they lack ontological axioms and assertions that are strong enough for deriving inconsistencies from. While all of the here mentioned approaches for erroneous identity links detection are useful in some cases, this paper presents a solution that can be applied to all data sets of the entire LOD Cloud.

In our previous work [14], we presented an identity graph of over half a billion owl:sameAs statements, obtained from a large crawl of the LOD Cloud, which includes links between resources from thousands of namespaces. We have also presented the deductive closure of this identity graph, which consists of over 35 billion implied identity statements. In other previous work [15], we presented a scalable approach for automatically assigning an error degree to each owl:sameAs statement in a given RDF graph, based on a community detection algorithm. In an extension of both these works, we show in this paper that even though the LOD Cloud contains erroneous owl:sameAs statements, it is still possible to use a very large subset of the LOD Cloud while at the same time minimizing the adverse effects of erroneous owl:sameAs statements. Specifically, we show that for the first time, Linked Data practitioners can now control in practice, the trade-off between (a) using more identity links, possibly not all correct, and benefiting from more contextual information from the LOD Cloud, and (b) using a smaller subset of correct identity links for limiting the risk of propagating erroneous identity links and information through the application of owl:sameAs semantics, i.e. transitive, symmetric, reflexive and property sharing. Since the choice of a certain subset must depend on a certain quality metric, we rely on the error degrees assigned in [15] as metric of identity links' quality, and on the approach presented in [14] for deducting new –higher quality– identity sets in a computationally efficient manner. As a first experiment on how Linked Data practitioners can benefit from such a trade-off, we generate in this work two new identity subgraphs, in which we compute their transitive closure, and analyze their new resulted identity sets. In the first identity subgraph, we adopt a more conservative approach for dealing with the identity problem at hand by considering only links of higher quality (i.e. with low error degree). In the second identity subgraph, we aim at limiting the risk of incorrect identity assertions while at the same time minimizing the loss of reachable information. This is done by discarding a smaller subset, consisting solely of highly questionable links (i.e. with high error degree). Both these subgraphs with their resulted identity sets after transitive closure are publicly available[①].

[①] https://zenodo.org/record/3345674.

The rest of this paper is structured as follows. Section 2 presents our approach for calculating the deductive closure of large identity graphs. Section 3 presents our approach for automatically assigning error degrees to owl:sameAs statements. Implementation and empirical evaluation of the deductive closure of a large identity graph obtained from the LOD Cloud of the two approaches, are presented in Section 4. Section 5 shows experimentally how the two approaches can be combined for selecting new, higher quality, identity subgraphs from the overall identity graph based on specified error degrees. Section 6 presents our conclusions on the conducted work.

2. EQUIVALENCE CLOSURE COMPUTATION

This section presents our approach for calculating and storing the deductive closure of a large collection of owl:sameAs statements (see [14] for more details). The problem at hand can be defined as follows: let N denote the set of RDF terms (i.e. IRIs, literals, and blank nodes) that appear in the subject or object position of at least one, non-reflexive, owl:sameAs triple. A *partitioning* of N is a collection of non-empty and mutually disjoint subsets $N_k \subseteq N$ (called partition members) that together cover N. In a network solely composed of N with their corresponding owl:sameAs edges, the connected components of this network are called *identity clusters*, and its partition members are called *identity sets*. According to the owl:sameAs semantics, all RDF terms belonging to the same identity set denotes the same real world entity: $\forall x, y \in N_k \rightarrow x = y$. In this work, we do not consider singleton identity sets, which are the result of terms that solely appear in reflexive owl:sameAs statements, and the result of terms which do not appear in any owl:sameAs statement.

In order to calculate the closure, each identity set should be closed under equivalence, while taking in consideration multiple dimensions of complexity:

The closure can be too large to store. We will later see that the LOD Cloud contains identity sets with cardinality well over 100K. It is not feasible to store the closure materialisation of each identity set, since the space consumption of that approach is quadratic in the size of the identity set (e.g. the closure of an identity set of 100K terms contains 10B identity statements). For this, we do not store the materialisation of the closure, but store the identity sets themselves, which is only linear in terms of the size of the universe of discourse (i.e. the set N of RDF terms).

$|N_k|$ can be too large to store. Even the number of elements within one identity set can be too large to store in memory. Since our calculation of the closure must have a low hardware footprint and must be future proof, we do not assume that every identity set is always small enough to fit in memory.

Data sets change over time. We calculate the identity closure for a large snapshot of the LOD Cloud. Since existing data sets in the LOD cloud are constantly changing, and new data sets are constantly added, our approach must support incremental updates of the closure. Specifically, our approach must allow for additions and deletions to be made over time, without requiring the entire closure to be recomputed.

Our approach is composed of the following steps: (1) extract the explicit owl:sameAs statements, (2) remove the owl:sameAs statements that are not necessary for calculating the deductive closure, and (3) compute the closure by partitioning the owl:sameAs network into several identity sets.

2.1 Explicit Identity Network

Given any RDF graph as input, e.g. the RDF merge of (a scrape of) the LOD Cloud, the first step of our approach is to extract the identity statements from this data graph (Definition 1).

Definition 1 (Data Graph). *A data graph is a directed and labelled graph $G = (V, E, \sum_E, l_E)$. V is the set of RDF terms and E is the set of node pairs or edges. $\sum_E$ is the set of edge labels. $l_E : E \rightarrow 2^{\sum_E}$ is a function that assigns to each edge $\in E$ a set of labels belonging to $\sum_E$ (with $l_E(e)$ representing the labels denoted to e).*

From a given data graph G, we can extract the explicit identity network G_{ex} (Definition 2), which is a directed labelled graph that only includes those edges whose labels include owl:sameAs.

Definition 2 (Explicit Identity Network). *Given a graph $G = (V, E, \sum_E, l_E)$, the corresponding explicit identity network $G_{ex} = (V_{ex}, E_{ex})$ is the edge-induced subgraph $G[\{e \in E \mid \{owl:sameAs\} \subseteq l_E(e)\}]$. V_{ex} is the set of RDF terms that appear in the subject and/or object position of at least one owl:sameAs statement ($V_{ex} \subseteq V$). E_{ex} is the set of node pairs or edges for which a statement $\langle x, owl:sameAs, y\rangle$ has been asserted in G ($E_{ex} \subseteq E$).*

2.2 Identity Network: Construction

Since identity is a reflexive, symmetric and transitive relation, the size of the explicit identity network can be significantly reduced prior to calculating the deductive closure. Specifically, we can reduce the size of G_{ex} into a more concisely represented undirected and labelled identity network I (Definition 3), without losing any information. Since reflexive owl:sameAs statements are implied by the semantics of identity, there is no need to represent them explicitly. In addition, since the symmetric statements e_{ij} and e_{ji} make the same assertion: that v_i and v_j refer to the same real-world entity, we can represent this more efficiently, by including only one undirected edge with a weight of 2. A weight of 1 is assigned for edges which either e_{ij} or e_{ji} are present in V_{ex}, but not both.

Definition 3 (Identity Network). *Given $G_{ex} = (V_{ex}, E_{ex})$, the identity network is an undirected labelled graph $I = (V_I, E_I, \{1, 2\}, w)$, where V_I is the set of RDF terms ($V_I \subseteq V_{ex}$), and E_I is the set of edges. $\{1, 2\}$ are the edges labels, and $w : E_I \rightarrow \{1, 2\}$ is the labelling function that assigns a weight w_{ij} to each edge e_{ij}. For an explicit identity network $G_{ex} = (V_{ex}, E_{ex})$, the corresponding identity network I is derived as follows:*

- $E_I := \{e_{ij} \in E_{ex} \mid i < j\}$
- $V_I := V_{ex}[E_I]$, *i.e. the vertex-induced subgraph*

- $w(e_{ij}) := \begin{cases} 2, & \text{if } e_{ij} \in E_{ex} \text{ and } e_{ji} \in E_{ex} \\ 1, & \text{if not} \end{cases}$

2.3 Identity Sets: Compaction and Closure

In this step, we can further reduce the input for the closure algorithm to a more concise set of ordered pairs. We call this preparation step *compaction*. Assuming a lexicographic order over RDF terms, we can reduce the input for the closure algorithm from an undirected labelled graph to a more concise set of pairs: $\{(x, y) \mid e_{x,y} \wedge x < y\}$. After compaction, we partition the terms V_I into different identity sets.

The partition of V_I into different identity sets consists of a mapping[2] from nodes to identity sets ($V_I \mapsto \mathcal{P}(V_I)$). For space efficiency, we store each identity set only once by associating a key with each identity set: $ID \mapsto_k \mathcal{P}(V_I)$; and map each RDF term to the key of the unique identity set that it belongs to $val : V_I \mapsto_v ID$. Hence, $val(x)$ gives us the identity set ID of an RDF term x, and the composition $key(val(x))$ gives us the identity set of x. For partitioning V_I into different identity sets, we use an incremental algorithm that parses each sorted identity pair (x, y), representing the output of the identity network compaction. The algorithm distinguishes between the following cases:

Case 1. Neither *x* nor *y* occurs in any identity set. A new identity set *id* is generated and assigned to both x and y: $x \mapsto_v id$; $y \mapsto_v id$; and $id \mapsto_k \{x, y\}$

Case 2. Only *x* already occurs in an identity set. In this case, the existing identity set of x is extended to contain y as well: $y \mapsto_v val(x)$; $val(x) \mapsto_k key\ (val(x)) \cup \{y\}$

Case 3. Only *y* already occurs in an identity set. Similar to the previous case.

Case 4. *x* and *y* already occur, but in different identity sets. In this case one of the two keys is chosen and assigned to represent the union of the two identity sets:

$$val(x) \mapsto_k key(val(x)) \cup key(val(y))$$
$$(\forall y' \in key(val(y)))(y' \mapsto_v val(x))$$

This is the costliest step, especially when both identity sets are large, but it is also relatively rare due to the sorting that was applied as part of the compaction step. A further speedup is obtained by choosing to merge the smaller of the two sets into the larger one instead of the other way round.

3. IDENTITY LINKS RANKING

In the previous section we presented our approach for calculating and storing the deductive closure of a large identity graph. In this section we present a scalable approach for automatically assigning an error degree for each owl:sameAs statement (see [15] for more details).

[2] Note that each term in V_I does indeed belong to a unique non-singleton identity set.

Our approach consists of applying an existing community detection algorithm to each identity cluster, i.e. connected component of *I*. Based on the detected communities in each identity cluster, an error degree is calculated and assigned for each owl:sameAs link. The error degree that is assigned to an owl:sameAs link depends on two factors: (a) the density of the community in which the two linked RDF terms reside, or the density of the interlinks between the two communities in case the linked RDF terms are not in the same community; and (b) whether the identity link is reciprocally asserted or not (i.e. has a weight of 2). This error degree is subsequently used for ranking identity links, allowing potentially erroneous links to be identified and potentially true links to be validated. We believe community detection to be a particularly good fit for scalable identity error detection, since it only requires the identity network itself as input. Existing approaches require additional inputs that are not always available (e.g. resource descriptions, property mappings, vocabulary alignments) and/or that do not always scale to the LOD Cloud (e.g. crowdsourcing). Also, the use of community detection for identity error detection does not require additional assumptions that may hold for some data sets, but not hold for others. For instance, the UNA is known not to hold for data sets that are constructed over a longer period of time and/or by a larger group of contributors. In order to be able to apply our approach on a large scale, including application to (a large scrape of) the LOD Cloud, we identify the following additional requirements for our algorithm:

Low memory footprint. Detection of erroneous identity links must not have a large memory footprint, since it must be able to scale to very large identity networks. In addition, we want our approach to be accessible to most researchers, by being able to run on a regular laptop.

Parallel computation. It must be possible to perform computation in parallel. With the increase of the number of cores, even on regular laptop and consumer devices, this will allow identity link error degrees to be computed more efficiently. Preferably, error degrees can be calculated immediately after publication of an owl:sameAs link in the LOD Cloud.

Updates. Calculation must be resilient against monotonic and non-monotonic updates. Since data are added to and removed from the LOD Cloud constantly, adding or removing an owl:sameAs statement must only require a re-ranking of the concerned links and must not require a complete recalculation.

Our approach is composed of the following main steps: (1) detect the community structure within each identity cluster, and (2) assign an error degree to each owl:sameAs link in the identity cluster. By relying on the identity network constructed in Section 2.2, and the resulted identity sets in Section 2.3, constructing the identity clusters is a straightforward phase.

3.1 Community Detection

Despite the absence of a universally agreed upon definition, communities are typically thought of as groups that have dense connections among their members, but sparse connections with the rest of the network. Community detection is a form of data analysis that seeks to automatically determine the community structure of a complex network, requiring solely information that is already encoded in

the network topology. Detecting a network's community structure is of great importance in many concrete applications and disciplines such as computer science, biology, and sociology, disciplines where systems are often represented as graphs. This has led to the emergence of several community detection algorithms, mostly making use of techniques from physics (e.g. spin model, optimization, random walks), as well as making use of computer science concepts and methods (e.g. non-linear dynamics, discrete mathematics) [16]. All such techniques aim at identifying group of nodes which are connected "more densely" to each other than to nodes in other groups. Hence, the differences between such methods ultimately come down to the precise definition of "more densely" and the algorithmic heuristic followed to identify such groups [17].

3.1.1 Community Detection Algorithms

Having a large number of community detection techniques, we relied on existing surveys for choosing the best performing community detection algorithm for our task. In their 2009 survey, Lancichinetti and Fortunato [18] carried out a comparative analysis of the performances of 12 community detection algorithms, that exploit some of the most interesting ideas and techniques that have been developed over the last years. The tests were performed against a class of benchmark graphs, with heterogeneous distributions of degree and community size, including the GN benchmark [19], the LFR benchmark [20, 21], and some random graphs. This study concludes that the modularity-based method by Blondel et al. [22], the statistical inference-based method by Rosvall and Bergstrom [23], and the multi-resolution method by Ronhovde and Nussinov [24] all have an excellent performance, with the additional advantage of low computational complexity. In a more recent study, Yang et al. [25] compared the results of 8 state-of-the-art community detection algorithms in terms of accuracy and computing time. Interestingly, only half of these algorithms were considered in the previous survey, with the tests also being conducted on the LFR benchmark. This study concludes that by taking both accuracy and computing time into account, the modularity-based method by Blondel et al. [22] outperforms the other algorithms. Given that [22] outperforms the other 15 algorithms in two different studies, with an additional advantage of low computational complexity, we deploy this algorithm for detecting the community structure in the owl:sameAs network. Next section presents an overview of this algorithm.

3.1.2 Louvain Algorithm

Given an identity cluster Q_k, the *Louvain* algorithm returns a set of non-overlapping communities $C(Q_k) = \{C_1, C_2, \ldots, C_n\}$ where:

- a community C of size $|C|$ (i.e. the number of nodes) is a subgraph of Q_k such that the nodes of C are densely connected (i.e. the modularity of the Q_k is maximized).
- $\cup_{1 \leq i \leq n} C_i = Q_k$ and $\forall C_i, C_j \in C(Q_k)$ *s.t.* $i \neq j$, $C_i \cap C_j = \emptyset$.

The Louvain algorithm is a method for detecting communities in large networks. It is a greedy non-deterministic method, introduced for the general case of weighted graphs, for the purpose of optimizing

the modularity of the partitions. The modularity of a partition is a scalar value between –1 and 1 that measures the density of links inside communities as compared to links between communities. In the case of weighted networks, modularity is defined as follows:

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j) \quad (1)$$

where:

A_{ij} represents the weight of the edge between the nodes *i* and *j*,

k_i **and** k_j represent the sum of the weights of the edges attached to the nodes *i* and *j*, respectively,

c_i **and** c_j represent the community to which to the nodes *i* and *j* are assigned, respectively,

$2m = \frac{1}{2}\sum_{i,j} A_{i,j}$ and representing the sum of all of the edge weights in the graph,

$\delta(u, v)$ is 1 if $u = v$ and 0 otherwise.

Modularity has been used to compare the quality of the partitions obtained by different methods, but also as an objective function to optimize [26]. Networks with high modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities. This is the intuition of the *Louvain* algorithm. Firstly, it starts out by assigning a different community to each node of a given network. Then, it moves each node over to one of its neighbour communities, specifically, neighbours which result in the highest contribution to the modularity measure. In the next step, each community from the previous step is regarded as a single node, and the same procedure is repeated until the modularity (which is always computed with respect to the original graph) no longer increases.

Although the exact computational complexity of the *Louvain* algorithm is not known, the method seems to run in time $O(N \log N)$, with N representing the number of nodes in the graph [22]. The exact modularity optimization is known to be NP-hard (non-deterministic polynomial-time hard), with most of the computational effort spent on the optimization at the first level.

3.2 Error Degree Computation

After detecting the community structure in each identity cluster, our approach assigns an error degree for each identity link based on its weight and the structure of the community(ies) in which its terms occur. We hypothesise that an identity link that is reciprocally asserted has a higher chance of being correct than a non-reciprocally asserted identity link. In addition, we hypothesise that not all resulting communities have the same quality. Based on these assumptions, we assign a lower error degree to owl:sameAs links that connect two terms that are within a densely connected community, or that connect two terms in two heavily interlinked communities. More precisely, to compute an error degree for each owl:sameAs link, we distinguish between two types of possible links: the intra- and inter-community links.

Definition 4 (Intra-Community Link). *Given a community C, an intra-community link in C noted by* e_C *is a weighted edge* e_{ij} *where* v_i *and* $v_j \in C$. *We denote by* E_C *the set of intra-community links in C.*

Definition 5 (Inter-Community Link). *Given two non overlapping communities* C_i *and* C_j, *an inter-community link between* C_i *and* C_j *noted by* $e_{C_{ij}}$ *is an edge* e_{ij} *where* $v_i \in C_i$ *and* $v_j \in C_j$. *We denote by* $E_{C_{ij}}$ *the set of inter-community links between* C_i *and* C_j.

For evaluating an *intra-community link*, we rely both on the density of the community containing the edge, and the weight of this edge. The lower the density of this community and the weight of an edge are, the higher the *error degree* will be.

Definition 6 (Intra-Community Link Error Degree). *Let* e_C *be an intra-community link of the community C, the intra-community error degree of* e_c *denoted by* $err(e_C)$ *is defined as follows:*

$$err(e_C) = \frac{1}{w(e_C)} \times \left(1 - \frac{W_C}{|C| \times (|C| - 1)}\right) \tag{2}$$

where $W_C = \sum_{e_C \in E_C} w(e_C)$.

For evaluating an *inter-community link*, we rely both on the density of the inter-community connections, and the weight of this edge. The less the two communities are connected to each other and the lower the weight of an edge is, the higher the *error degree* will be.

Definition 7 (Inter-Community Link Error Degree). *Let* $e_{C_{ij}}$ *be an inter-community link of the communities* C_i *and* C_j, *the inter-community error degree of* $e_{C_{ij}}$ *denoted by* $err(e_{Cij})$ *is defined as follows:*

$$err(e_{C_{ij}}) = \frac{1}{w(e_{C_{ij}})} \times \left(1 - \frac{W_{C_{ij}}}{2 \times |C_i| \times |C_j|}\right) \tag{3}$$

where $W_{C_{ij}} = \sum_{e_{C_{ij}} \in E_{C_{ij}}} w(e_{C_{ij}})$.

4. IMPLEMENTATION AND EXPERIMENTS

In the previous two sections, we presented our approach for computing and storing the identity graph and its deductive closure, as well as our approach for assigning an error degree for each owl:sameAs link. In this section, we describe how these two approaches can be combined and used in practice to deal with the Semantic Web identity problem. The overall workflow of the identity network extraction, compaction and closure is displayed in Figure 1.

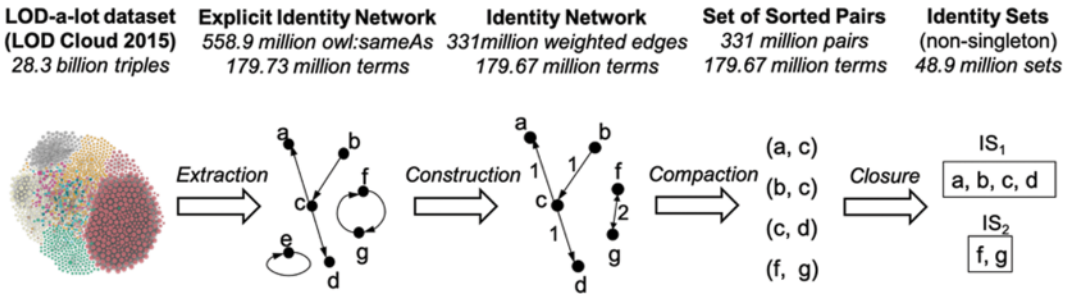

Figure 1. Workflow of the identity network extraction, compaction and closure.

4.1 Data Graph

We conduct our experiments on the *LOD-a-lot* data set [27], but any crawl of the LOD Cloud can be used as the starting point of our approach. We refer to this initial large-scale crawl of the LOD Cloud as our data graph (Definition 1). *LOD-a-lot* is the graph merge of the data obtained by the large-scale crawling and cleaning infrastructure of the LOD Laundromat [28]. The result of this merge is published in a single, publicly accessible Header Dictionary Triples (HDT) file that is 524 GB in size. This data graph contains over 28.3 billion unique triples, over 5 billion unique terms, that are related by over 1.1 billion unique predicates.

4.2 Explicit Identity Network Extraction

We use the *LOD-a-lot* HDT Dump to extract the explicit identity network (G_{ex}), and the HDT C++ library[③] to stream the result set of the following SPARQL query to a file. This process takes around 27 minutes:

```
select distinct ?s ?p ?o {
bind (owl:sameAs AS ?p)
?s ?p ?o }
```

The results of this query are unique (keyword distinct) and the projection (?s ?p ?o) returns triples instead of pairs, so that regular RDF tools for storage and querying can be used. The explicit identity statements are stored in the order in which they are asserted by the original data publishers. This SPARQL query returns more than 558.9 million triples that connect 179.73 million terms. These owl:sameAs triples are written to an N-Triples file, which is subsequently converted to an HDT file. The HDT creation process takes almost four hours using a single CPU core. The resulting HDT file is 4.5 GB in size, plus an additional 2.2 GB for the index file that is automatically generated upon first use. This large collection of identity statements can be queried[④] directly from the browser or downloaded as HDT[⑤].

③ https://github.com/rdfhdt/hdt-cpp.
④ https://krr.triply.cc/krr/sameas/graphs or http://sage.univ-nantes.fr/sparql/sameAs.
⑤ https://zenodo.org/record/1973099.

4.3 Identity Network Construction

From the extracted G_{ex}, we build the identity network (Definition 3) containing around 331M weighted edges and 179.67M terms. As a result, we leave out around 2.8M reflexive edges and around 225M duplicate symmetric edges. We also leave out 67,261 nodes that only appear in such removed edges.

4.4 Identity Sets Compaction and Closure

The identity network can be reduced to a set of sorted pairs prior to calculating the closure. For this we use GNU sort which takes less than 35 minutes on an SSD disk. Then, we compute the identity closure that consists of a map from nodes to identity sets. In order to build an efficient implementation of this key-value scheme, we need a solution that (i) uses almost no memory but is allowed to use a (SSD) disk, (ii) is able to store billions of key-value pairs, and (iii) allows such pairs to be added/removed dynamically over time. For this we use the RocksDB[⑥] persistent key-value store through a SWI Prolog API[⑦] that was designed for this purpose, allowing to simultaneously read from and write to the database. Since changes to the identity relation can be applied incrementally, the initial creation step only needs to be performed once. The computation of the closure takes just under 5 hours using 2 CPU cores on a regular laptop. The result is a 9.3GB on-disk RocksDB database: 2.7GB for mapping each term to an identity set ID ($V_I \mapsto_v ID$), and 6.6GB for mapping each identity set ID to its corresponding identity set ($ID \mapsto_k P(V_I)$). These files are publicly available[⑧].

The number of non-singleton identity sets is 48,999,148. The *LOD-a-lot* file, from which we construct I, contains 5,093,948,017 unique terms. This means that there are 5,044,948,869 singleton identity sets in the deductive closure. Figure 2 shows that the size distribution of non-singleton identity sets is very uneven and fits a power law with exponent 3.3 ± 0.04. Around 64% of non-singleton identity sets (31,337,556 sets) contain only two terms. There are relatively few large identity sets, with the largest one containing 177,794 terms. By looking at the IRIs of this identity set, we can observe that it contains a large number of terms denoting different countries, cities, things and persons (e.g. Bolivia, Dublin, Coca-Cola, Albert Einstein, the empty string, etc.). This observation shows the need of detecting the erroneous owl:sameAs links which led to such false equivalences after transitive closure.

⑥ https://rocksdb.org.
⑦ https://github.com/JanWielemaker/rocksdb.
⑧ https://zenodo.org/record/3345674.

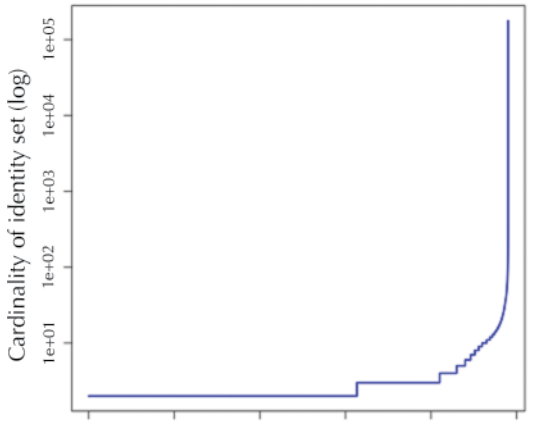

Figure 2. The distribution of identity set cardinality in G_{im}. The *x*-axis lists all 48,999,148 non-singleton identity sets.

4.5 Links Ranking

Relying on the constructed identity network *I* and the resulting 48,999,148 identity sets in the deductive closure, we reconstruct the identity clusters of *I*. These identity clusters represent a partitioning of *I* into connected components. Then, we apply the *Louvain* algorithm for detecting the community structure in each identity cluster. As discussed in Section 3.1.2, the *Louvain* method is a greedy and non-deterministic algorithm. Meaning that in different runs, the algorithm might produce different communities, with no insurance that the global maximum of modularity will be attained. For this, we have run *Louvain* 10 times on each identity cluster, and finally considered the community structure with the highest modularity. After detecting the communities, we assign an error degree to all edges of each identity cluster.

4.5.1 Quantitative Evaluation

This process of community detection and error degree computation took 80 minutes, resulting an error degree to each owl:sameAs statement in the identity network (around 556M statements after discarding the reflexive ones). The error degree distribution of these statements is presented in Figure 3. This figure shows that around 73% of the statements have an error degree below 0.4, whilst around 5% of the owl:sameAs statements have an error degree higher than 0.8. Whilst this distribution is mainly caused by the high number of symmetrical identity statements in the LOD Cloud, it also indicates that most identity clusters have a rather dense structure. The 179.67M terms of the identity network were assigned into a total of 55.6M communities, with the communities' size varying between 2 and 4,934 terms (averaging 3 terms per community). The JAVA implementation[⑨] of the ranking process, and the computed error degrees[⑩] for all owl:sameAs statements are publicly available.

⑨ http://github.com/raadjoe/LOD-Community-Detection.
⑩ https://sameas.cc.

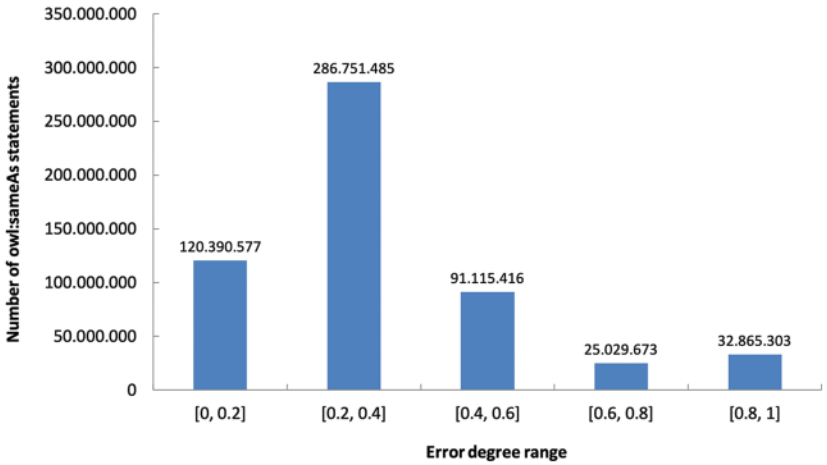

Figure 3. Error degree distribution of all owl:sameAs statements in the LOD-a-lot.

4.5.2 Qualitative Evaluation

In order to evaluate the accuracy of our ranking approach, we conducted several manual evaluations. This evaluation is an extension of the ones conducted in [15], in terms of the number of manually evaluated links and the followed analyses.

In this evaluation, we aim at defining a threshold x of the error degree, in which owl:sameAs links that have an error degree $\leq x$ have high probability of correctness, and links which have an error degree $>x$ have high probability of being erroneous. In order to determine this threshold, four of the paper's authors were asked to evaluate a number of owl:sameAs links. The judges relied on the descriptions[®] associated to the terms in the *LOD-a-lot* data set [27], and did not have any prior knowledge about each link's error degree (i.e. whether they are evaluating a well-ranked link or not). In order to avoid any incoherence between the evaluations, the judges were asked to justify all their evaluations, and were given the following instructions: **(a) the same:** if two terms denote the same entity (e.g. *Obama* and the *First Black US President*), **(b) related:** not intended to refer to the same entity but closely related (e.g. *Obama* and the *Obama Administration*, or *Obama* and the *Wikipedia article of Obama*), **(c) unrelated:** not the same nor closely related (e.g. *Obama* and the *Indian Ocean*), **(d) can't tell:** in case there are no sufficient descriptions available for determining the meaning of both terms (i.e. non-dereferenced IRIs and appearing solely as subjects or objects of owl:sameAs statements in the LOD Cloud). Based on the judges' evaluations, we deploy the following terms:

True Positives (TP) referring to owl:sameAs links which have an error degree $>x$ and were evaluated by the judges as incorrect links (related or unrelated).

® Judges were asked to not consider the owl:sameAs statements associated to a term.

False Positives (FP) referring to owl:sameAs links which have an error degree >x and were evaluated by the judges as true identity links.

True Negatives (TN) referring to owl:sameAs links which have an error degree ≤x and were evaluated by the judges as true identity links.

False Negatives (FN) referring to owl:sameAs links which have an error degree ≤x and were evaluated by the judges as incorrect links (related or unrelated).

An approach of erroneous link detection can be evaluated using the classic evaluation measures of precision, recall and accuracy defined as follows:

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

We consider that when a Semantic Web expert is not able to confirm the correctness of a certain identity link due to the absence of necessary descriptions for one of the two involved terms, an automated approach will have the same limitations. Following this assumption, we will not consider links judged by the experts as "can't tell" in the accuracy, precision, and recall evaluations.

Firstly the judges were asked to evaluate 200 owl:sameAs statements (50 each), representing a sample of each bin of the error degree distribution shown in Figure 3. The main goal of this first evaluation of a small set of links, is to test whether the defined error degree is indeed an indicator for correctness (i.e. whether the probability of finding erroneous identity links increases with the increase of the error degree). From the results presented in Table 1, we can observe the following:

Table 1. Evaluation of 200 owl:sameAs links, with each 40 links randomly chosen from a certain range of error degree. The percentages between parentheses are calculated without considering the links evaluated as "can't tell".

Error degree range	0–0.2	0.2–0.4	0.4–0.6	0.6–0.8	0.8–1	**Total**
same	35 (100%)	22 (100%)	18 (85.7%)	7 (77.8%)	15 (68.2%)	**97 (89%)**
related + unrelated	0 (0%)	0 (0%)	3 (14.3%)	2 (22.2%)	7 (31.8%)	**12 (11%)**
related	0	0	2	2	2	**6**
unrelated	0	0	1	0	5	**6**
can't tell	5	18	19	31	18	**91**
Total	**40**	**40**	**40**	**40**	**40**	**200**

- The higher an error degree is, the more likely that the link is erroneous.
- 100% of the evaluated links with an error degree ≤ 0.4 are correct.

- When the error degree is between 0.4 and 0.8, 83.3% of the owl:sameAs links are correct, whilst in 13.3% of the cases, such links might have been used to refer to two different, but related terms.
- An owl:sameAs with an error degree > 0.8 is a less reliable identity statement, referring in 31.8% of the cases to two different terms.

By taking the highest available threshold in this table ($x = 0.8$), the evaluation suggests a precision of 31.8% ($\frac{7}{15+7}$), and an accuracy of 81.6% ($\frac{35+22+18+7+7}{35+22+21+9+22}$). This suggests that a higher threshold is required in order to detect erroneous owl:sameAs statements with higher precision. Therefore, we increased the threshold to 0.99, and manually evaluated additional samples in order to better evaluate the accuracy and the precision of our approach. The 200 owl:sameAs statements evaluated in Table 1 represents our first sample. We briefly describe in the following the three additional samples, and their purposes:

Sample 2. It represents a set of 60 owl:sameAs statements, with error degrees between 0.9 and 1, randomly chosen from identity clusters of various sizes. We mention that 21 of these owl:sameAs were judged as "can't tell" by the judges.

Sample 3. In 2013, the authors of [13] manually evaluated 95 owl:sameAs statements linking Freebase with DBpedia terms, and they were all judged as correct. It is the only publicly available external gold standard from the LOD Cloud. Out of these 95 links, only 78 were found in our data graph. Verifying their error degrees, only one link was assigned an error degree higher than 0.99 (FP), with the 77 other links having an error degree ranging from 0.52 to 0.94 (TN).

Sample 4. In order to evaluate the recall of our approach, we have verified how our approach can rank newly introduced erroneous owl:sameAs statements. Firstly, we have chosen 40 random terms from the identity network, making sure that all these terms are different and not explicitly owl:sameAs[©] (e.g. dbr:Paris, dbr:Strawberry, dbr:Facebook). From the 40 selected terms, we have generated all the possible 780 undirected edges between them. We added separately, each edge e_{ij} to the identity network with $w(e_{ij}) = 1$, calculated its error degree, and removed it from the identity network before adding the next one. The resulted error degrees of the newly introduced erroneous identity links range from 0.87 to 1. When the threshold is fixed at 0.99, the recall of detecting erroneous identity links is 93%, with 725 (TP) out of the 780 added links (TP+FN) having an error degree > 0.99.

The evaluation of these owl:sameAs samples is presented in Table 2, with a threshold of 0.99. The results presented in this table suggests that our approach can classify an identity link (as correct or erroneous) with a 91.9% accuracy when the threshold is set at 0.99. We are aware that accuracy is sometimes misleading in imbalanced data sets, as it fails to detect an approach which is biased towards the dominated class of the data set. However, in the case of the four samples we considered, we believe that accuracy can be a

© But be sure to include some terms that currently belong to the same identity set.

good estimation on how this automatic approach matches human judgement, since it performs well on samples dominated by correct owl:sameAs (e.g. sample 3), and samples dominated by erroneous links (e.g. sample 4).

Table 2. Accuracy of the approach on the four manually evaluated samples, based on a threshold of 0.99. Links evaluated as "can't tell" by the judges are discarded.

Sample	TN	TP	FN	FP	Total	Accuracy
Sample 1	97	0	12	0	**109**	**88.9%**
Sample 2	6	20	5	8	**39**	**66.6%**
Sample 3	77	0	0	1	**78**	**98.7%**
Sample 4	0	725	55	0	**780**	**92.9%**
Total	**180**	**745**	**72**	**9**	**1006**	**91.9%**

5. FIXING OWL:SAMEAS IN THE LOD CLOUD

This section combines our approach for computing the transitive closure of owl:sameAs (Section 2), with our approach that assesses the quality of owl:sameAs (Section 3). Specifically, we use the error degrees computed in Section 4.5 for generating new, higher quality, identity networks and sets in the LOD Cloud. This section makes the following contributions:

- Presents two new identity networks, with the first identity network discarding "potentially erroneous" owl:sameAs, and the second identity network containing only "high quality" owl:sameAs.
- Presents the transitive closure of the two newly constructed identity networks.
- Presents a use case and a benchmark for evaluating the advantages and the limitations of using these new identity networks and their deducted identity sets.

5.1 Link Selection Criteria

In order to classify an owl:sameAs as "potentially erroneous" or of "high quality", we rely on the manually evaluated links previously described in Table 1 and Table 2. Specifically, Table 1 shows that 100% of the manually evaluated links with an error degree ≤ 0.4 are correct (57/57). Hence, suggesting that links with such error degrees can be classified as "probably correct". In addition, the manual evaluation presented in Table 2, shows that our approach has high accuracy when the threshold is fixed at 0.99 (~92% accuracy). Hence, it also suggests that identity links with an error degree > 0.99 can be classified, with high degree of confidence, as "potentially erroneous". In order to put these two hypotheses to the test, we construct two new subgraphs of the identity network with the following characteristics:

1. All owl:sameAs statements in the identity network with an error degree > 0.99 are discarded.
2. All owl:sameAs statements in the identity network with an error degree > 0.4 are discarded.

In the following, we present a quantitative and qualitative evaluation of these two identity subgraphs. In addition, we generate and evaluate their corresponding identity sets, and discuss their impact. These new subgpraphs, with their deducted identity sets, are publicly available[®].

5.2 Quantitative Evaluation

Figure 4 presents a comparison in terms of the number of identity pairs, terms and resulting identity sets between the closure of (a) the original identity network containing all 331M identical pairs[®] in the LOD Cloud; (b) a subset of the identity network containing only links with an error degree ≤ 0.99; and (c) a subset of the identity network containing only links with an error degree ≤ 0.4. Firstly, this figure shows that the number of RDF terms that belong to non-singleton identity sets decreases from 179.67M in the original closure to 179.34M in the closure. This indicates that about 330K RDF terms are solely involved in "potentially erroneous" owl:sameAs statements. In addition, this evaluation shows that discarding the about 500K pairs with an error degree > 0.99 results in about 75K additional non-singleton identity sets. These sets are the result of the partitioning of several large –probably incorrect– identity sets. On the other hand, considering only the 205K "highly probable" identical pairs in the LOD Cloud (i.e. linked by an owl:sameAs with error degree ≤ 0.4) decrease the number of the resulting non-singleton identity sets by more than 70% (from 48.9M non-singleton identity sets in the original closure to 15.9M). This result is expected since only 53M out of the 179M terms in the LOD Cloud (about 30%) are involved in identity statements with such high confidence.

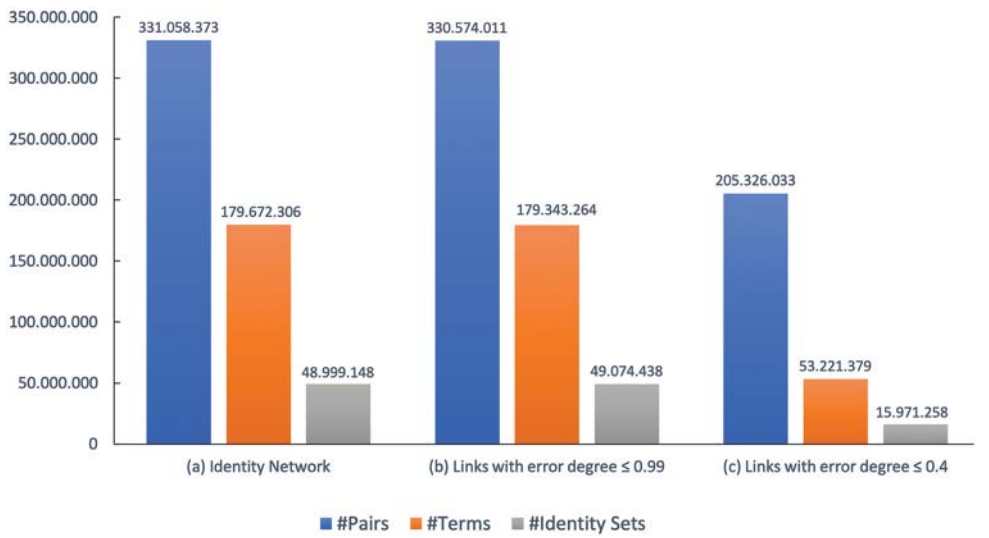

Figure 4. Comparison of the original identity network and its transitive closure, with the two newly constructed identity subgraphs.

® https://zenodo.org/record/3345674.

® Note that in the identity network, reflexive and duplicate symmetric links are discarded.

In order to analyze closely the impact of these new computed closures, we focus firstly on the largest identity set. This identity set includes originally 177,794 terms, connected by 2.8M edges that are the result of the compaction of 5.5M owl:sameAs (about 1% of the total number of owl:sameAs). This identity set includes terms referring to a large number of different real-world entities, falsefully claimed to be the same (in an explicit or implicit manner). When owl:sameAs links, with error degree >0.99 are discarded (respectively discarding links with error degree >0.4), the terms of this identity set are partitioned into 1,086 non-singleton identity sets (resp. 640 sets), with an average of 160 terms per set (resp. 50 terms). The largest of these derived non-singleton identity sets contains 3,686 terms (resp. 157 terms) and the smallest set in each of the two closures contains only two terms. As a result of discarding links with an error degree >0.99, we find that 4,146 terms, originally belonging to the largest identity set (2.3% of the terms), now appear in singleton identity sets. On the other hand, discarding links with an error degree >0.4, results in 145,934 terms originally belonging in the largest identity set (82% of the terms) to be present in singleton identity sets (i.e. not identical to any other term anymore).

Finally, in terms of computational efficiency, the two new deductive closures are much faster to compute. Compared to the original closure that takes about 5 hours using two CPU cores on a regular laptop, the closure (b) and (c) takes about 2.5 hours and about 1 hour, respectively. This significant difference in computation time between the closure (a) and (b), despite the relatively small difference in the number of included pairs (about 500K pairs), is consistent with our observation in Section 2.3, stating that the computation time for calculating the deductive closure is dominated by the time that is needed to merge large identity sets.

5.3 Qualitative Evaluation

Until now, we computed two additional equivalence closures, based on discarding owl:sameAs with certain error degrees. We also compared these new subgraphs to the original identity graph in terms of the number of resulting identity sets, number of terms included in non-singleton identity sets, and the computation time for deducting their identity sets. In this section, we analyze the quality of these subraphs, and compare it to the quality of the original identity graph (i.e. all the owl:sameAs links of the LOD Cloud). For conducting this qualitative analysis, we look closely at the specific use case of the identity cluster containing the DBpedia term of the former US president Barack Obama (dbr:Barack_Obama). This identity cluster, presented in Figure 5, includes 440 terms connected by 7,615 edges. These edges are the result of the compaction of 14,917 owl:sameAs statements. Applying the *Louvain* algorithm on this identity cluster results in four non-overlapping communities[®], presented in Figure 6.

® https://www.sameas.cc/explicit/obama-class.svg.

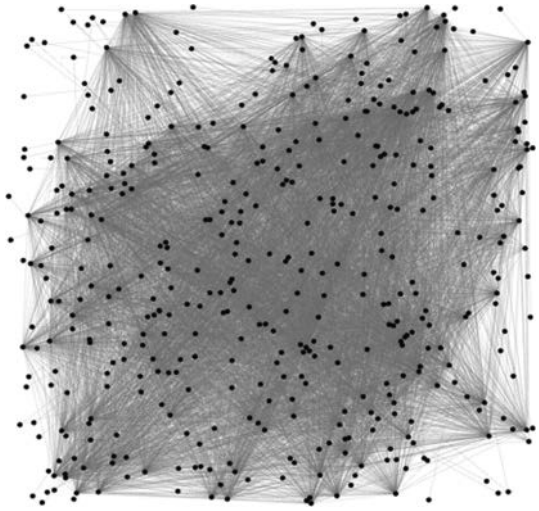

Figure 5. "Barack Obama" identity cluster.

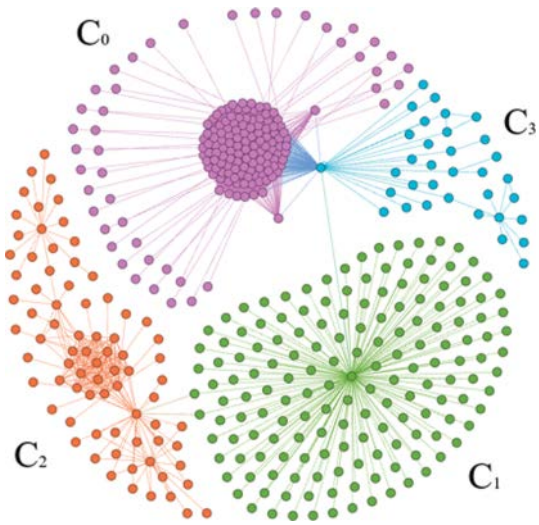

Figure 6. Community structure of the "Barack Obama" identity cluster.

In order to analyze the impact of discarding links with certain error degrees, the authors were asked to manually annotate each RDF term of this identity set. By dereferencing the IRIs, and looking at their descriptions in the *LOD-a-lot* data set, the authors were able to annotate 338 out of the 440 RDF terms. These 338 terms were judged to refer to 8 different real world entities (e.g. Barack Obama the person, his presidency, his senate career). The other 102 RDF terms do not have enough descriptions to be confidently annotated by the judges, and will be discarded for the rest of this evaluation. Now that all the terms in this identity set are annotated, we conduct two complementary evaluations. The first evaluation conducted on

the links' level (Section 5.3.1), allows us to evaluate how accurately our approach can classify each identity link, based on different thresholds. The second evaluation conducted on the closure level (Section 5.3.2), allows us to evaluate the impact of discarding few identity links on the quality of the overall closure. Both the code and the data necessary for replicating these analyses are publicly available[®].

5.3.1 "Barack Obama" Identity Cluster – Links Evaluation

The identity cluster containing the DBpedia term of the former US president "Barack Obama" contains 14,917 owl:sameAs, between 440 RDF terms. After discarding the 102 terms that could not be manually annotated and their corresponding links[®], this identity cluster now contains 14,613 owl:sameAs linking 338 RDF terms. Based on our manual annotation of these terms, we find that there are 26 owl:sameAs that link two RDF terms referring to different entities (0.17% of the links are erroneous). This evaluation shows that even though this identity cluster mostly contains correct owl:sameAs links, the presence of a small number of erroneous ones have led to the false equivalence of RDF terms referring to 8 different real world entities. Table 3 presents the evaluation of our approach in classifying the owl:sameAs links of this identity cluster according to both considered thresholds 0.4 and 0.99. From this table, we can observe that there are only two owl:sameAs links with an error degree > 0.99, with both these links being indeed erroneous (TP=2 and FP=0 in closure *b*). Hence, these results suggest a 100% precision (2/2), and a 7.69% recall (2/26) for our approach when the threshold is fixed at 0.99. Conversely, when the threshold is set at 0.4, the recall of our approach increases to 100% (i.e. all 26 erroneous links in this identity cluster have an error degree >0.4), with the precision decreasing to 3.16% (due to the additional 795 flagged links that are actually correct). For both thresholds, the accuracy is considerably high, which is expected in the case of an imbalanced data set.

Table 3. Precision, recall and accuracy, based on two thresholds (0.99 and 0.4) for the Barack Obama identity set. Links evaluated as "can't tell" by the judges are discarded.

Closure	Error degree threshold	TN	TP	FN	FP	Total	Recall	Precision	Accuracy
(b)	0.99	14,587	2	24	0	14,613	7.69 %	100 %	99.83 %
(c)	0.4	13,792	26	0	795	14,613	100 %	3.16 %	94.55 %

5.3.2 "Barack Obama" Identity Cluster – Closure Evaluation

In the previous evaluation, we showed that the identity cluster of "Barack Obama" in the LOD Cloud contains 26 erroneous owl:sameAs that led to the false equivalence of RDF terms referring to 8 different real world entities. The evaluation in Table 3 shows that depending on the chosen threshold, a Linked Data practitioner can decide whether to remove only few erroneous owl:sameAs without removing any correct ones (i.e. closure b), or remove all erroneous owl:sameAs but discard at the same time a large number of

® https://github.com/raadjoe/obama-lod-identity-analysis.

® Note that these terms were not discarded during the computation of the error degree, but only discarded in the evaluation.

correct ones (i.e. closure c). In this section, we evaluate the impact of both approaches on the overall closure, and compare it to the original closure of all owl:sameAs links in the *LOD-a-lot* data set. Table 4 presents the differences in the resulted identity sets between the three closures, where we can observe the following:

Gold Standard. Our manual annotation shows that the 338 RDF terms in the original identity cluster refer to 8 different real world entities. Ideally we expect that these terms to be partitioned into 8 different identity sets $\{GS_1, \ldots, GS_8\}$, with GS_1 containing the 260 RDF terms referring to the person Barack Obama, GS_2 containing the 47 terms referring to his presidency, and so forth.

Closure (a). The original closure of all owl:sameAs links in the *LOD-a-lot* data set results in all these 338 RDF terms to belong in one identity set A_1. These 338 terms are explicitly connected by 14,613 owl:sameAs links, and result in 56,953 distinct pairs.

Closure (b). After discarding the two owl:sameAs links with an error degree >0.99, the closure of the 14,611 remaining owl:sameAs links partitions these 338 RDF terms into two non-singleton identity sets B_1 and B_2. The former contains 270 RDF terms and the latter contains the remaining 68 terms. Table 4 shows that by solely discarding two erroneous owl:sameAs links from this identity cluster, our approach was able to separate all terms referring to Obama's presidency and presidential transition® from the rest of the terms. However, despite this significant quality improvement, both identity sets are still logically inconsistent (i.e. contain terms that refer to different real world entities).

Closure (c). After discarding the 821 owl:sameAs links with error degree >0.4, the closure of the remaining 13,792 correct identity links partitions the 338 RDF terms into 219 identity sets. Out of these identity sets, only the identity set C_1 is non-singleton, containing 120 RDF terms that refer to the person Barack Obama. Although this closure results in a large number of identity sets (compared to the expected 8 sets), the resulting closure is now semantically correct.

® terms referring to the period in which Obama won the US election in November 2008 until his inauguration in January 2009 when his presidency started.

Table 4. Comparison of the original identity network closure, the closure (b) and (c), with the Gold Standard.

		Closure (a)	Closure (b)		Closure (c)			
	Real World Entity	A_1	B_1	B_2	C_1	C_2	...	C_{219}
GS_1	*Barack Obama*	260	260	0	120	1		0
GS_2	*Obama's Presidency*	47	0	47	0	0		0
GS_3	*Obama's Presidential Transition*	22	1	21	0	0		0
GS_4	*Obama's Senate Career*	5	5	0	0	0		0
GS_5	*Obama's Presidential Centre*	1	1	0	0	0		0
GS_6	*Obama's Biography*	1	1	0	0	0		0
GS_7	*Obama's Photos*	1	1	0	0	0		0
GS_8	*Black President*	1	1	0	0	0		1
	# Terms in Identity Set	**338**	**270**	**68**	**120**	**1**	**1**	**1**
	# Identity Sets	**1**	**2**		**219**			
	# Explicit Identity Links	**14,613**	**14,611**		**13,792**			
	# Distinct Pairs	**56,953**	**38,593**		**7,140**			

In order to further evaluate the quality of these three closures, we also rely on the classic evaluation measures of precision, recall and accuracy. However, since the evaluation in this case is conducted on the level of the resulting identity clusters, we re-define the notion of True/False Positives/Negatives for evaluating our approach in terms of the number of remaining correct/erroneous distinct pairs (instead of explicit owl:sameAs). Hence, all possible 56,953 distinct pairs X between the 338 manually annotated RDF terms represent the universe of discourse in the following evaluation.

True Positives (TP). Pairs $(a,b) \in X$, in which a and b are in the same identity cluster in both the gold standard and in the resulting closure.

False Positives (FP). Pairs $(a,b) \in X$, in which a and b are in the same identity cluster in the resulting closure but in different clusters in the gold standard.

True Negatives (TN). Pairs $(a,b) \in X$, in which a and b are in different clusters in both the gold standard and the resulting closure.

False Negatives (FN). Pairs $(a,b) \in X$, in which a and b are in the same identity cluster in the gold standard, but in different clusters in the closure.

Table 5 presents the recall, precision and accuracy evaluation of each closure. For the closure (a), it is expected to have a 100% recall since all the 338 considered terms in this evaluation belong originally to

the same identity cluster. Since ideally, these terms should be partitioned into 8 different clusters, this table shows that the original closure of all owl:sameAs links contains 21,961 erroneous pairs, resulting in a 61% precision of the original closure. On the other hand, for the closure (c) it is also expected to have a 100% precision since all pairs (*a*,*b*) that belong to the same resulting identity cluster are indeed identical (TP). This result can also be observed in Table 4, where all identity sets resulting from the closure (c) are either singleton, either in the case of C_1 refer solely to the person Obama. However, since it partitions the 338 RDF terms into 219 identity clusters (where the ideal partition is 8 clusters), the recall of this closure is significantly lower (20%), due to the large number of correct pairs that have been separated in this closure (FN). Finally, the closure (b) manages to have both a high recall (99.9%) and high precision (90.6%). This is due to the large number of non-identical pairs that have been separated (TN), and the large number of identical pairs that were kept in the same cluster (TP) after discarding the only two owl:sameAs with an error degree >0.99. For instance, we can see from Table 4 that all 260 terms referring to the person Obama are still in the same cluster B_1, and at the same time separated from all 47 terms referring to his presidency.

Table 5. Precision, recall and accuracy evaluation of the three closures.

Closure	TN	TP	FN	FP	Total	Recall	Precision	Accuracy
(a)	0	34,992	0	21,961	56,953	100 %	61.4 %	61.4 %
(b)	18,339	34,971	21	3,622	56,953	99.9 %	90.6 %	93.6 %
(c)	21,961	7,140	27,852	0	56,953	20 %	100 %	51 %

6. CONCLUSION

In this paper we showed that even though a global knowledge graph like the LOD Cloud may contain incorrect identity statements, it is still possible to use subgraphs of this knowledge graph in a productive way. This is done by extracting all owl:sameAs links, computing their transitive closure, and assigning an error degree to each link based on the identity cluster's community structure. Since the here presented approach only takes a couple of hours to be computed over a large scrape of the LOD Cloud (over 500 million explicit and 35 billion implicit owl:sameAs statements) while using a regular laptop, this trade-off between identity subgraph size and link validity can actually be used in practice. In fact, our qualitative evaluation on a benchmark of about 1K owl:sameAs links, and a manually annotated identity cluster containing about 14K owl:sameAs links suggests that discarding a small number of links, with an error degree higher than 0.99 (about 0.17% of the links), can significantly enhance the quality of the resulting closure (correctness of the closure increases from 61% to 93%). Our results also suggest that discarding a larger number of links, with an error degree higher than 0.4 (about 27% of the links) can lead to semantically valid identity clusters (i.e. which do not contain two terms referring to different real world entities). These results could be further improved by combining or comparing results from multiple community detection methods.

Finally, based on the level of correctness that is required within a given application, a data practitioner may choose a different error degree, resulting in a smaller/larger subgraph depending on whether erroneous identity statements are considered more/less acceptable. In addition, some applications might choose to requalify links with certain error degrees to a less strict notion of identity, such as the one encoded in skos:closeMatch. This will allow practitioners to benefit from the presence of these asserted links, while reducing the semantic inconsistencies when reasoning is applied. We believe that automated and scalable data selection approaches, like the one presented in this paper, will see uptake over time, as they allow data practitioners to extract maximal value from the LOD Cloud, and thereby make the Semantic Web succeed as an integrated and reliable knowledge space.

AUTHOR CONTRIBUTIONS

J. Raad (j.raad@vu.nl) was co-responsible for the design of the research, the implementation of the approach, the analysis of the results, and the writing of the manuscript. W. Beek (wouter@triply.cc) was co-responsible for the design of the research, the implementation of the approach, the analysis of the results, and the writing of the manuscript. F. van Harmelen (frank.van.harmelen@vu.nl) has contributed to the design of the research, the analysis of the results, and the writing of the manuscript. N. Pernelle (nathalie.pernelle@lri.fr) has contributed to the the design of the research, the analysis of the results, and the writing of the manuscript. F. Saïs (fatiha.sais@lri.fr) has contributed to the the design of the research, the analysis of the results, and the writing of the manuscript. J. Wielemaker (j.wielemaker@vu.nl) has contributed to the implementation of the approach.

REFERENCES

[1] A. Hogan, A. Zimmermann, J. Umbrich, A. Polleres & S. Decker. Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. Web Semantics: Science, Services and Agents on the World Wide Web 10(2012), 76–110. doi:10.1016/j.websem.2011.11.002.

[2] J. Raad. Identity management in knowledge graphs. PhD dissertation, University of Paris-Saclay, 2018. Available at: https://tel.archives-ouvertes.fr/tel-02073961.

[3] H. Halpin, P.J. Hayes, J.P. McCusker, D.L. McGuinness & H.S. Thompson. When owl:sameAs isn't the same: An analysis of identity in Linked Data. In: International Semantic Web Conference, 2010, pp. 305–320. doi:10.1007/978-3-642-17746-0_20.

[4] J. Raad, N. Pernelle, F. Saïs, W. Beek & F. van Harmelen. The sameas problem: A survey on identity management in the web of data. arXiv preprint. arXiv:1907.10528, 2019.

[5] P. Bouquet, H. Stoermer & D. Giacomuzzi. Okkam: Enabling a web of entities. In: CEUR Workshop Proceedings, 2007, pp. 1–8. Available at: http://www.ceur-ws.org/Vol-249/submission_150.pdf.

[6] H. Glaser, A. Jaffri & I. Millard. Managing co-reference on the semantic web. In: WWW2009 Workshop: Linked Data on the Web LDOW, 2009, pp. 1–6. Available at: http://www.ceur-ws.org/Vol-538/ldow2009_paper11.pdf.

[7] M. Mountantonakis & Y. Tzitzikas. On measuring the lattice of commonalities among several linked data sets. In: Proceedings of the VLDB Endowment, 2016, pp. 1101–1112. doi:10.14778/2994509.2994527.

[8] J. Cuzzola, E. Bagheri & J. Jovanovic. Filtering inaccurate entity co-references on the linked open data. In: International DEXA Conference, 2015, pp. 128–143. doi: 10.1007/978-3-319-22849-5_10.
[9] G. de Melo. Not quite the same: Identity constraints for the web of linked data. In: The Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013, pp. 1092–1098. Available at: http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/download/6313/6872.
[10] A. Valdestilhas, T. Soru & A.C.N. Ngomo. Cedal: Time-efficient detection of erroneous links in large-scale link repositories. In: International Conference on Web Intelligence, 2017, pp. 106–113. doi: 10.1145/3106426.3106497.
[11] L. Papaleo, N. Pernelle, F. Saïs & C. Dumont. Logical detection of invalid sameas statements in RDF data. In: International Conference EKAW, 2014, pp. 373–384. doi: 10.1007/978-3-319-13704-9_29.
[12] C. Guéret, P. Groth, C. Stadler & J. Lehmann. Assessing linked data mappings using network measures. In: Extended Semantic Web Conference, 2012, pp. 87–102. doi: 10.1007/978-3-642-30284-8_13.
[13] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer & J. Lehmann. Crowdsourcing linked data quality assessment. In: International Semantic Web Conference, 2013, pp. 260–276. doi:10.1007/978-3-642-41338-4_17.
[14] W. Beek, J. Raad, J. Wielemaker & F. van Harmelen. sameas. cc: The closure of 500m owl: sameas statements. In: Extended Semantic Web Conference, 2018, pp. 65–80. doi:10.1007/978-3-319-93417-4_5.
[15] J. Raad, W. Beek, F. van Harmelen, N. Pernelle & F. Saïs. Detecting erroneous identity links on the web using network metrics. In: International Semantic Web Conference, 2018, pp. 391–407. doi: 10.1007/978-3-030-00671-6_23.
[16] S. Fortunato. Community detection in graphs. Physics Reports 486(2010), 75–174. doi:10.1016/j.physrep.2009.11.002.
[17] M.A. Porter, J.P. Onnela & P.J. Mucha. Communities in networks. Notices of the AMS 56(2009), 1082–1097. doi: 10.1016/j.cnsns.2012.03.023.
[18] A. Lancichinetti & S. Fortunato. Community detection algorithms: A comparative analysis. Physical Review E 80(2009), 056117. doi:10.1103/PhysRevE.80.056117.
[19] M. Girvan & M.E. Newman. Community structure in social and biological networks. In: Proceedings of the national academy of sciences, 2002, pp. 7821–7826. doi:10.1073/pnas.122653799.
[20] A. Lancichinetti, S. Fortunato & F. Radicchi. Benchmark graphs for testing community detection algorithms. Physical Review E 78(2008), 046110. doi:10.1103/PhysRevE.78.046110.
[21] A. Lancichinetti & S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E 80(2009), 016118. doi:10.1103/PhysRevE.80.016118.
[22] V.D. Blondel, J. Guillaume, R. Lambiotte & E. Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 10(2008), 10008. doi:10.1088/1742-5468/2008/10/P10008.
[23] M. Rosvall & C.T. Bergstrom. Maps of random walks on complex networks reveal community structure. In: Proceedings of the National Academy of Sciences, 2008, pp. 1118–1123. doi: 10.1073/pnas.0706851105.
[24] P. Ronhovde & Z. Nussinov. Multiresolution community detection for megascale networks by information-based replica correlations. Physical Review E 80(2009), 016109. doi: 10.1103/PhysRevE.80.016109.
[25] Z. Yang, R. Algesheimer & C.J. Tessone. A comparative analysis of community detection algorithms on artificial networks. Scientific Reports 6(2016), 30750. doi: 10.1038/srep30750.
[26] M.E.J. Newman & M. Girvan. Finding and evaluating community structure in networks. Physical Review E 69(2004), 026113. doi:10.1103/PhysRevE.69.026113.

[27] J.D. Fernández, W. Beek, M.A. Martínez-Prieto & M. Arias. Lod-a-lot. In: International Semantic Web Conference, 2017, pp. 75–83. doi:10.1007/978-3-319-68204-4_7.

[28] W. Beek, L. Rietveld, H. R. Bazoobandi, J. Wielemaker & S. Schlobach. Lod laundromat: A uniform way of publishing other people's dirty data. In: International Semantic Web Conference, 2014, pp. 213–228. doi:10.1007/978-3-319-11964-9_14.

AUTHOR BIOGRAPHY

Joe Raad is a post-doctoral researcher in the Knowledge Representation & Reasoning group at the Vrije Universiteit Amsterdam, The Netherlands. His research interests focus on identity management in Knowledge Representation systems, large-scale empirical study of semantics, and semantic technologies deployment for Digital Humanities. As part of his research, he has co-developed sameAs.cc and MetaLink.
ORCID: 0000-0002-7891-7738

Wouter Beek is a post-doctoral researcher in the Knowledge Representation & Reasoning group at the Vrije Universiteit Amsterdam and co-founder of Triply. He is interested in the Semantic Web as a platform for knowledge-intensive applications, the deployment of large-scale knowledge bases for innovative reuse, and the interaction between Web semantics and pragmatics, including the empirical study of semantics. As part of his research, he has co-developed the LOD Laundromat, LOD Search, and sameAs.cc.
ORCID: 0000-0003-0250-9655

Frank van Harmelen is a Professor in Knowledge Representation & Reasoning in the Computer Science department (Faculty of Science) at the Vrije Universiteit Amsterdam, The Netherlands. Since 2000, he has played a leading role in the development of the Semantic Web. He was co-PI on the first European Semantic Web project (OnToKnowledge, 1999), which laid the foundations for the Web Ontology Language OWL. OWL has become a worldwide standard, it is in wide commercial use, and it has become the basis for an entire research community. He co-authored the *Semantic Web Primer*, the first academic textbook of the field and now in its third edition, which is in worldwide use (translations in 5 languages, and 10,000 copies sold of the English edition alone). He was one of the architects of Sesame, an RDF storage and retrieval engine, which is in wide academic and industrial use with over 200,000 downloads. This work received the 10-year impact award at the 11th International Semantic Web Conference in 2012, which is the most prestigious award in the field. In recent years, he pioneered the development of large-scale reasoning engines. He was scientific director of the 10m euro EU-funded Large Knowledge Collider, a platform for distributed computation over semantic graphs with billions of edges. The prize-winning work with his student Jacopo Urbani has improved the state of the art by two orders of magnitude. He is scientific director of The Network Institute. In this interdisciplinary research institute some 150 researchers from the Faculties of Social Science, Humanities and Computer Science collaborate on research topics in computational Social Science and e-Humanities. He is a fellow of the European AI Society ECCAI (membership limited to 3% of all European AI researchers), in 2014, he was admitted as member of the Academia Europaea (limited to the top 5% of researchers in each field), and in 2015 he was admitted as Member of the Royal Netherlands Society of Sciences and Humanities (450 members across all sciences).

ORCID: 0000-0002-7913-0048

Jan Wielemaker is a senior researcher at the Vrije Universiteit Amsterdam (VUA) and the Centrum voor Wiskundeen Informatica (CWI, The Netherlands). He is the lead developer of SWI-Prolog and responsible for the development and maintenance of the linked data infrastructure libraries for SWI-Prolog.
ORCID: 0000-0001-5574-5673

Nathalie Pernelle is a Professor of Computer Science, member of the LIPN (Laboratoired' Informatique de Paris Nord), at the University of Paris in France, where she moved from Paris-Saclay University in 2019. Her research interests are related to knowledge discovery in data graphs. She has in particular studied models and algorithms for data linking, rule mining, and for the semantic annotation of unstructured documents. She has been involved in many academic and industrial projects related to various domains such as biological or geographical data, bibliographical knowledge bases, asbestos diagnoses, or problems related to the General Data Protection Regulation.
ORCID: 0000-0003-1487-393X

Fatiha Saïs is currently an associate Professor - HDR at the Computer Science Research Laboratory (LRI) of Paris Saclay University, France. She is currently the co-head of LaHDAK group (Large-scale Heterogeneous Data and Knowledge). Her research focuses on: identity management in the Web of data; knowledge graph fusion; knowledge discovery from RDF graphs; and more recently on the veracity assessment in knowledge graphs. Her work has been included in more than 20 national, industrial and European research projects. She has published more than 60 research papers in national and international conferences and journals like, ISWC (International Semantic Web Conference), *Journal of Web Semantics* and *Journal of Data Semantics*. She served as a PC member for international conferences (ECAI, ESWC, K-Cap, ICCS, etc.), national conferences (EGC, IC, BDA) and organized and chaired several national and intentional workshops and conferences (WebToTouch, EGC, Verita, SoWedo, JDSE, etc.).
ORCID: 0000-0002-6995-2785